به نام هستی بخش

سلام بر مهدی که انتظارش را فقط دل عاشق،

که ترنم هر باران بهاری و هر روز نبی امید میکشد...

پاسخ تمرین 6 (فصل های 8، 9)

درس پایگاه داده‌ها، بهار 1403

---

۱) اگر ترتیب record dataها مشابه یا نزدیک به ترتیب data entry ها باشد شاخص ما clustered است در غیراین صورت unclustered است. هزینه بازیابی رکوردهای داده‌ای، بسیار به clustered بودن یا نبودن داده‌های بستگی دارد.

- در پرس‌وجو از نوع equality search تفاوتی در هزینه پرس‌وجو با clustered کردن شاخص نمی‌توان ایجاد کرد و لذا در اینجا تفاوتی بین این دو نوع شاخص وجود ندارد.
- در پرس‌وجو از نوع range search استفاده از شاخص clustered به شکل قابل توجهی هزینه را کاهش می‌دهد، اما توجه شود که قبلاً یک شاخص clustered تعریف شده و فقط یک شاخص از این نوع می‌توانیم داشته باشیم، پس شاخص ما لزوماً از نوع unclustered می‌تواند باشد.

۲)

۱- در این روش data entryها را بر روی کلید جست‌وجو هش می‌کنیم. در این حالت داده‌هایی که مقدار کلید جست‌وجو یکسانی دارند را می‌توانیم به‌سرعت پیدا کنیم. در این روش، هر ایندکس مجموعه‌ای از باکت‌ها است و با اعمال تابع هش مشخص می‌شود که data entry به کدام باکت تعلق دارد. این نوع شاخص برای equality search بسیار خوب عمل می‌کند، زیرا داده‌ها با کلید جست‌وجو یکسان کنار هم قرار می‌گیرند، اما برای range search مناسب نمی‌باشد.

۲- هر دو شامل data entryهایی هستند که به data recordها اشاره می‌کنند، در ۳ alternative لیست ridها نگهداری می‌شود، ۳ alternative جای کمتری می‌گیرد ولی می‌تواند باعث به‌وجودآمدن data entryها با سایز متفاوت شود (نیازی به اشاره یا ترسیم مورد آخر برای نمره کامل نیست). فقط یک شاخص از نوع alt.1 می‌توانیم داشته باشیم و در غیر این صورت data duplication خواهیم داشت. از سایر شاخص‌ها به هر تعداد می‌توانیم داشته باشیم.

۳- بله. توجه شود که در این حالت با هر به‌روزرسانی، هزینه به‌روزرسانی شاخص‌ها نیز اضافه می‌شود. همچنین ممکن است استفاده از خود شاخص نیز به ما کمکی نکند و در این صورت باید هزینه استفاده از شاخص را نیز بپردازیم. (ارائه هر مثال صحیحی قابل‌قبول است.)

۳)

۱- در روش لیست پیوندی از دو لیست پیوندی دوطرفه برای نگهداری صفحات پر و خالی استفاده می‌کنیم. هرگاه داده‌ای از صفحه‌های پر حذف شد و صفحه جای خالی داشت، آن را به صفحات لیست دیگر انتقال می‌دهیم و هرگاه هنگام نوشتن صفحه‌ای پر شد، آن را به لیست صفحات پر منتقل می‌کنیم. در روش page directory، تعدادی directory را به‌صورت لیست پیوندی

قرار می‌دهیم و در هرکدام اطلاعات مربوط به صفحات (اشاره‌گر به آن صفحه و جاداشتن یا نداشتن) را می‌نویسیم. همچنین می‌توان میزان فضای خالی یک صفحه را نیز ذخیره کنیم.

۲-

الف) در بدترین حالت به $5\ IO$ احتیاج خواهیم داشت:

Read page dir 1 + Read page dir 2 + read data page + write data page + write on page dir2

ب) بدترین حالت این است که فقط در ششمین صفحه دارای فضای خالی رکورد مربوطه بتواند قرار بگیرد و پس از آن این صفحه پر شود و لازم باید به قسمت صفحات پر منتقل شود:

Read header page + 6*(read empty pages) + write page6 + read a full page + 3*(pointer update) = 12 IO

توجه شود که باید اشاره‌گرها در صفحات پر نیز آپدیت شوند و لذا یکی از صفحات پر نیز باید خوانده شود.

۴)

۱- در حالت index-only اطلاعات موجود در data entryها برای پاسخ‌دادن به کوئری‌ها کافی است و نیازی به واکشی خود داده‌ها از حافظه نیست. استفاده از این روش باعث کاهش هزینه پرس‌وجوها می‌شود.

۲- a) شاخص باید به شکل <category, rate, price> باشد. در این حالت data entryها در ابتدا بر اساس category مرتب شده‌اند و آن‌هایی که category یکسان دارند بر اساس rate مرتب شده‌اند پس به‌راحتی می‌توان آن‌هایی که ۵> rate دارند را حذف کرد. چون در نهایت کمترین price خواسته شده پس باید price نیز در شاخص باشد.

b) شاخص باید به شکل <category, price> باشد.

۳- شرایطی برای استفاده از index-only-planها نیاز است، از جمله پوشش کامل ستون‌ها، تمام ستون‌های موردنیاز پرس‌وجو باید در اندیس موجود باشند و یک شاخص مناسب بر روی آن‌ها ایجاد شده باشد. اگر این شرایط لازم برقرار نباشند، پرس‌وجوی مطرح شده نیاز به دسترسی به پایگاه‌داده پیدا خواهد کرد (توجه شود برخی عملیات همچون UPDATE و DELETE صرف‌نظر از ایندکس نیاز به دسترسی به پایگاه‌داده و تغییر در آن دارند) دیگر نمی‌توان از index-only-plan استفاده کرد.

۵) به اسلایدهای درس مراجعه شود.

۶)

Requested Page	MRU	Hit?	LRU	Hit?
1	1 x x x	0	1 x x x	0
3	1 3 x x	0	1 3 x x	0
2	1 3 2 x	0	1 3 2 x	0
5	1 3 2 5	0	1 3 2 5	0
6	1 3 2 6	0	6 3 2 5	0
4	1 3 2 4	0	6 4 2 5	0
5	1 3 2 5	0	6 4 2 5	1
3	1 3 2 5	1	6 4 3 5	0
2	1 3 2 5	1	6 4 3 2	0
1	1 3 2 5	1	1 4 3 2	0
6	6 3 2 5	0	1 6 3 2	0
1	1 3 2 5	0	1 6 3 2	1
2	1 3 2 5	1	1 6 3 2	1
1	1 3 2 5	1	1 6 3 2	1
3	1 3 2 5	1	1 6 3 2	1
2	1 3 2 5	1	1 6 3 2	1
4	1 3 4 5	0	1 4 3 2	0
1	1 3 4 5	1	1 4 3 2	1
2	2 3 4 5	0	1 4 3 2	1
5	2 3 4 5	1	1 4 5 2	0
Hit Rate:		9/20 = 45%		8/20 = 40%

۷)

Clustered hash index:

- equality search: $2 * D$

با یک D به bucket موردنظر می‌رسیم و rid را به دست آورده و سپس با یک D دیگر صفحه مربوط به آن data record را به دست می‌آوریم.

- range search: $D * B$

این شاخص کمکی نمی‌کند پس باید همه data recordها را بررسی کنیم.

Clustered tree index:

- equality search: $(log_F 0.25 * B + 1) * D$

می‌توانیم با $log_F 0.25$ به rid مدنظر برسیم چرا که data entry، اندازه ۱۰٪ از فضای data record است. و فقط ۴۰٪ صفحه ها پر هستند و $\frac{10}{100} * \frac{100}{40} = \frac{1}{4} = 0.25$ و با یک D می‌توانیم رکورد را بخوانیم.

- range search: $(log_F 0.25 * B + match\ pages) * D$

با $log_F 0.25$ به rid اولین data entry در بازه می‌رسیم. حالا به تعداد صفحات مربوط به rid بقیه داده ها جلو می‌رویم.

Unclustered tree index:

- equality search: $(log_F 0.25 * B + 1) * D$

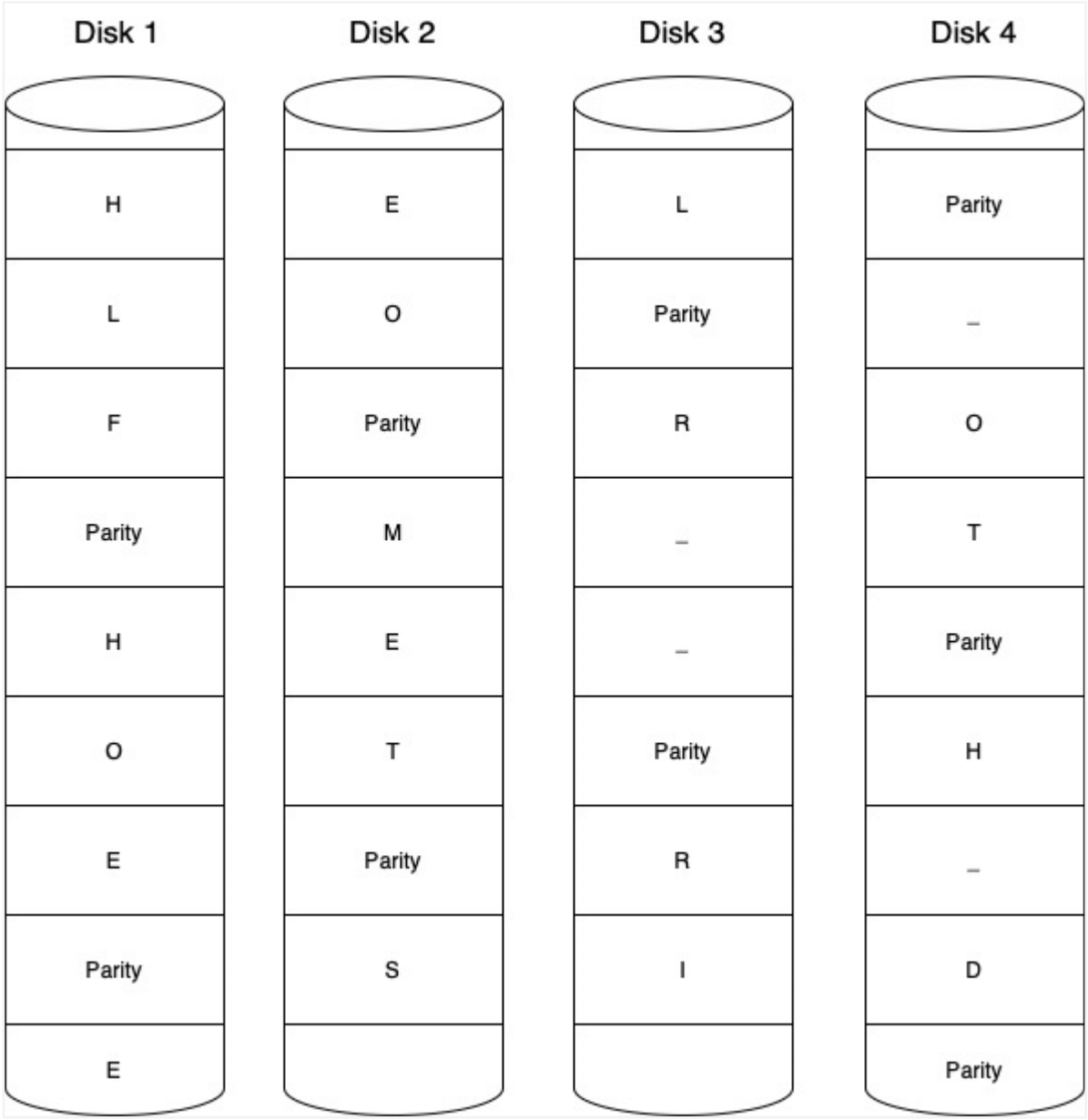
همانند Clustered tree index است.

- range search: $(log_F 0.25 * B + matchrecords + matchpages) * D$

همانند Clustered tree index می‌باشد؛ ولی با این تفاوت که به‌ازای هر data record نیز صفحه‌ای می‌خوانیم.

۸) ۱- به اسلایدهای درس مراجعه شود.

۲-



سربلند و پیروز باشید.